Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXXVI, págs. 1173-1181 (artículo completo)
José G. Etse, Bibiana M. Luccioni, Martín A. Pucheta, Mario A. Storti (Eds.)
San Miguel de Tucumán, 6-9 Noviembre 2018

HETEROGENEOUS POROUS MEDIA SIMULATION

Nicolás I. Dazeo, Javier A. Dottori, Gustavo A. Boroni and Ignacio Larrabide

Pladema, Universidad Nacional del Centro de la Provincia de Buenos Aires, Pinto 399, 7000 Tandil, Argentina, ndazeo@pladema.exa.unicen.edu.ar, http://www.pladema.net

Keywords: CFD, Porous Media, Finite Volume Method, OpenFOAM

Abstract. Intracranial aneurysms are vascular disorders in which weakness in the wall of a cerebral artery or vein causes a localized dilation of the blood vessel. Flow diversion is an endovascular technique where a flow diverter stent is placed in the parent blood vessel to divert blood flow away from the aneurysm itself. Simulation by computational fluid dynamics is an attractive method to study flow diverters, particularly to model the small gaps between stent struts as a porous media. In many cases obstructions are not equal across the free medium and the porous one must be heterogeneous. Finite Volume Method solves numerical problems of computational fluid dynamics, splitting the region of interest in cells of small volumes. Porous media are usually modeled as a set of simulation cells described in a dictionary with constant porosity parameters (Homogeneous medium). An heterogeneous medium can be described as multiple homogeneous media, one by one. However, creating multiple homogeneous porous media is a tedious job if each simulation cell requires different parameters. Also, porous medium sets creates overheads on memory and processor load. The open source tool OpenFOAM is a open source C++ toolbox for field operations and partial differential equations solving using Finite Volume Method, including computational fluid dynamics. The tool is well prepared to describe heterogeneous fields. In this work, porous media coefficients are described as tensor fields. A steady state flow solver considering this fields is developed. The fidelity of the solver is then studied qualitatively and quantitatively.

1 INTRODUCTION

Flow Diverters (FDs) stents are endovascular devices used to treat intracraneal aneurysms. They are placed in the parent vessel of an aneurysm to divert the flow from the aneurysm sac (Sadasivan et al., 2009; Ma et al., 2012). Computational Fluid Dynamics (CFD) has been largely used to predict the behavior and assess changes in local blood flow when a FD is placed on the parent vessel (Mut et al., 2014; Larrabide et al., 2013, 2012). The default way for modeling stent wires is as no-slip boundaries inside the fluid domain.

FD permeability and porosity are the main features driving their ability to modify local flow (Kim et al., 2008). Because of this, it seems natural to model such devices using Porous Medium (PM) models (Dazeo et al., 2017). In the case of braided FD, porosity is determined not only by the FDs design, but also by the final positioning of the braided wire mesh inside the vessel.

The braided mesh pores of the uncompressed regions span angles of $90°$ between wires, while in areas of local compression the angle ratio change up to $15°$ and $165°$ reaching an occlusion. Most authors use homogeneous PM to model the braided mesh but, because of the local variations, an heterogeneous model can lead to a better approximation.

To model the blood flow patterns using heterogeneous PM, the open source tool OpenFOAM is used. This tool is a toolbox designed for field operations and partial differential equation solving using FV, including CFD (Jasak et al., 2007). The tool provides a PM solver were the user defines a porous region and constant Darcy and Fochheimer values. The solver iterates over each volume cell in the porous region, applying a volumetric force. This implementation use same coefficients for all porous cells in a zone. Therefore, to model an heterogeneous porous media multiple zones must be defined. But each zone requires multiple dictionaries, with the consequent memory overhead: A dictionary needs around 500 characters per cell versus 72 bytes (9 floats of 4 bytes for 2 tensors) per cell for both tensor fields. Moreover, iteration over zones is not parallelizable, since zone overlapping is allowed. In this work, a steady state solver capable of simulate non-uniform PM modeling porosity coefficients as a tensor field is implemented in OpenFOAM and then validated.

2 MATERIALS AND METHODS

2.1 Finite Volume Software

The open source Finite Volume toolkit OpenFOAM in it's version 4.1 was used to develop the solver. The toolkit use operator overloading that allows algorithms to be expressed in a natural way. In this case, a steady state solver was implemented based on the SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithm.

2.2 Porosity Fields

It is required to define the two variables D and F, representing Darcy and Forchheimer fields, respectively, at the cell center of the mesh. The Listing 1 shows how this variables are implemented in OpenFOAM. Both objects are tensor fields, which are the size of the mesh. There is a Darcy and a Forchheimer tensor value for each volume cell.

```
Info<< "Reading field D\n" << endl;
volTensorField D
(
    IOobject
    (
```

```
            "D",
            runTime.timeName(),
            mesh,
            IOobject::MUST_READ,
            IOobject::AUTO_WRITE
        ),
        mesh
);

Info<< "Reading field F\n" << endl;
volTensorField F
(
        IOobject
        (
            "F",
            runTime.timeName(),
            mesh,
            IOobject::MUST_READ,
            IOobject::AUTO_WRITE
        ),
        mesh
);
```

Listing 1: Field construction on createFields.H

2.3 Porosity Equation

For the required application, Navier-Stokes equations for incompressible flows must be solved. Particularly, in the region of the porous medium, a Brinkman-Forchheimer volumetric force must be applied. The momentum equation is then as follows:

$$\rho\left(\mathbf{U}\cdot\nabla\right)\mathbf{U} = -\nabla P + \mu\nabla^2\mathbf{U} - \left(\mu\cdot\mathbf{D} + {}^1\!/_2\rho\cdot\mathbf{U}\cdot\mathbf{F}\right)\cdot\mathbf{U}, \tag{1}$$

where $\mathbf{U}$ is the velocity, P is the pressure, $\mathbf{D}$ is the Darcy tensor, $\mathbf{F}$ is the Forchheimer tensor, and μ, and ρ are the viscosity and density of the fluid.

The Equation 1 implemented in OpenFOAM code is depicted in Listing 2. The fvm::div operator for example, takes the convective flux phi as a coefficient field defined over the faces of the control elements and U as the variable field defined over the cell centroids, and returns a system of equations including a Left Hand Side (LHS) matrix and a Right Hand Side (RHS) source. These LHS matrices and RHS vectors are generated for each of the operators and then added or subtracted as needed to yield the final system of equations that represent the discretized set of algebraic equations defined over each element of the computational domain Moukalled et al. (2016).

```
        tmp<fvVectorMatrix> tUEqn
    (
        fvm::div(phi, U)
      + MRF.DDt(U)
      + turbulence->divDevReff(U)
```

```
 ==
     fvOptions(U)
);
fvVectorMatrix& UEqn = tUEqn.ref();
```

Listing 2: Incompresible Navier-stokes momentum equation in OpenFOAM

The PM terms are added to the momentum equation in Listing 3 . In the code, the diagonal elements of the LHS matrix are obtained with `UEqn.A()`. Volumetric forces caused by the PM are added there.

```
tmp<volTensorField> tTU = tensor(I)*UEqn.A();

volTensorField & AU = tTU.ref();
AU = AU + mu*D + (rho*mag(U))*F/2.0;
AU.correctBoundaryConditions();
```

Listing 3: Porosity terms added as volumetric forces.

With the constructed momentum equation, new values for the velocity field are approximated, as depicted in Listing 4. The matrix is inverted to move it to the RHS of the equation. Then the velocity field on the LHS is iteratively approximated, because the RHS also depends on it. The `&` operator does the dot product, and `UEqn.H()` returns all terms on the right-hand side, excluding those involving pressure.

```
trTU = inv(tTU());
volVectorField gradp(fvc::grad(p));

for (int UCorr=0; UCorr<nUCorr; UCorr++)
{
    U = trTU() & (UEqn.H() - gradp);
}
U.correctBoundaryConditions();
```

Listing 4: Implicit approximation of velocity field from momentum equation.

SIMPLE algorithm is implemented for pressure velocity coupling. The equation for pressure is solved in Listing 5.

```
tmp<volVectorField> tHbyA = constrainHbyA(trTU()&UEqn.H(), U, p);

volVectorField& HbyA = tHbyA.ref();

tUEqn.clear();
surfaceScalarField phiHbyA("phiHbyA", fvc::flux(HbyA));

MRF.makeRelative(phiHbyA);

adjustPhi(phiHbyA, U, p);

while (simple.correctNonOrthogonal())
{
```

```
        tmp<fvScalarMatrix> tpEqn;

        tpEqn = (fvm::laplacian(trTU(), p) == fvc::div(phiHbyA));

        fvScalarMatrix pEqn = tpEqn.ref();

        pEqn.setReference(pRefCell, pRefValue);

        pEqn.solve();

        if (simple.finalNonOrthogonalIter())
        {
            phi = phiHbyA - pEqn.flux();
        }
    }

    p.relax();

    U = HbyA - (trTU()&fvc::grad(p));

    U.correctBoundaryConditions();
```

Listing 5: The equation for pressure solved in PEqn.H.

The velocity at the face is obtained by interpolating the semi-discretized form of momentum equation. The semi discretized momentum equation does not include the pressure gradient term. The continuity equation along with semi-discretized momentum equation are used to solve the pressure. The equation for pressure is solved for prescribed number of non orthogonal corrector steps. Then the flux is corrected based on the solved pressure. The pressure is under-relaxed for momentum corrector and the velocity is corrected.

2.4 Validation

An ideal squared section channel of $1 \times 10^{-3}\mathrm{m}$ side, and $1 \times 10^{-2}\mathrm{m}$ long was used. The channel is divided in finite volumes cells of $2.5 \times 10^{-5}\mathrm{m}$. The finite volumes at distances of $5 \times 10^{-4}\mathrm{m}$, $2 \times 10^{-4}\mathrm{m}$, $1 \times 10^{-4}\mathrm{m}$ from the center of the PM were incrementally refined. Boundaries were set with a $0.2\,^{\mathrm{m}}\!/\!_{\mathrm{s}}$ velocity condition at the inlet, a $0\mathrm{Pa}$ pressure outlet, and slip lateral walls. Also, the simulation of a FD stent over a real geometry was realized.

3 RESULTS

The solver was tested in ideal geometries for validation against analytic values and a homogeneous solver when possible. Also, the flow in the parent vessel of an aneurysm with a FD modeled as a PM is solved. The flow in the vessel is then qualitatively compared to a DNS simulation.

3.1 Homogeneous PM

An homogeneous PM was placed in the center of the channel. The medium covers the whole section of the channel with a length ΔL of $5 \times 10^{-5}\mathrm{m}$. The Darcy and Forchheimer fields in the medium where set to constant values chosen from a typical FD stent of 1643133150 and

894.3699319 respectively, and both 0 outside it. The pressure gradient is different from zero only within the porous medium, which generates a pressure drop ΔP between both sides of the channel. This pressure drop can be obtained from the Equation 1:

$$\Delta P = (\mu \cdot D + 1/2 \rho \cdot U \cdot F) \cdot \Delta L \cdot U, \tag{2}$$

$$0.062906215 = (3.77 \times 10^{-6} \cdot 1643133150 + 1/2 \cdot 0.2 \cdot 894.3699319) \cdot 5 \times 10^{-5} \cdot 0.2, \tag{3}$$

and it was also measured in both simulations.

Method	Pressure drop (Pa/ρ)
Analytic	0.062906215
porousFieldSimpleFoam	0.0628532
porousSimpleFoam	0.0628532

Table 1: Pressure drop expected from analytic equations and simulation results.

The Table 1 shows expected values obtained from equations, the pressure drop measured from the implemented solver (porousFieldSimpleFoam) and the one measured from a homogeneous solver provided by OpenFOAM (porousSimpleFoam). Both solvers got the same result of $0.0628532(\mathrm{Pa}/\rho)$. The solvers had a relative error of 0.084%.

3.2 Gradient PM

The main purpose of the application is to solve fluid equations in interaction with heterogeneous porous media. To evaluate heterogeneous capabilities, a non-uniform PM was placed in the center of the channel. The medium covers the whole section of the channel with a length ΔL of $5 \times 10^{-5}\mathrm{m}$. The porosity fields in the PM were set as a function of the position (x) in the channel:

$$D(x) = D_0 \cdot (x - x_0), \tag{4}$$

$$F(x) = F_0 \cdot (x - x_0), \tag{5}$$

where D_0 and F_0 where set to 3283133150 and 1800 respectively, and x_0 is the position x where the medium starts. This pressure drop can be obtained similar to the homogeneous media, from the Equation 1:

$$\Delta P = (\mu \cdot D + 1/2 \rho \cdot U \cdot F) \cdot \Delta L^2 \cdot U \tag{6}$$

$$3.14 \times 10^{-6} = (3.77 \times 10^{-6} \cdot 3283133150 + 1/2 \cdot 0.2 \cdot 1800) 1/2 \cdot 2.5 \times 10^{-9} \cdot 0.2 \tag{7}$$

Method	Pressure drop (Pa/ρ)
Analytic	0.000003143
porousFieldSimpleFoam	0.00000314264

Table 2: Analytic pressure drop and simulation result.

The Table 2 shows expected values obtained from Equation 6 and the pressure drop measured from the implemented solver. The solver got a pressure drop of $0.00000314264(\mathrm{Pa}/\rho)$, and a relative error of 0.0114%.

3.3 Real Geometry

The solver will be applied for blood flows interacting with FD stents in intracraneal aneurysms. Hence a test in a real geometry was done. On a patient specific geometry, a braided mesh was placed on the parent vessel across the aneurysm neck estimated with the fast virtual deployment method Larrabide et al. (2012). Porosity coefficients for each braided cell where estimated using the method presented by Raschi et al. (2014). The FD is then modeled as a porous region of same thickness as the struts. Also, the same configuration is simulated using no-slip boundary conditions to model the FD.

Simulation results can be compared the Figure 1 The PM creates a pressure drop between the parent vessel and the aneurysm sac as the boundary conditions method. Although the pressure drop of the FD is overestimated by the PM model.

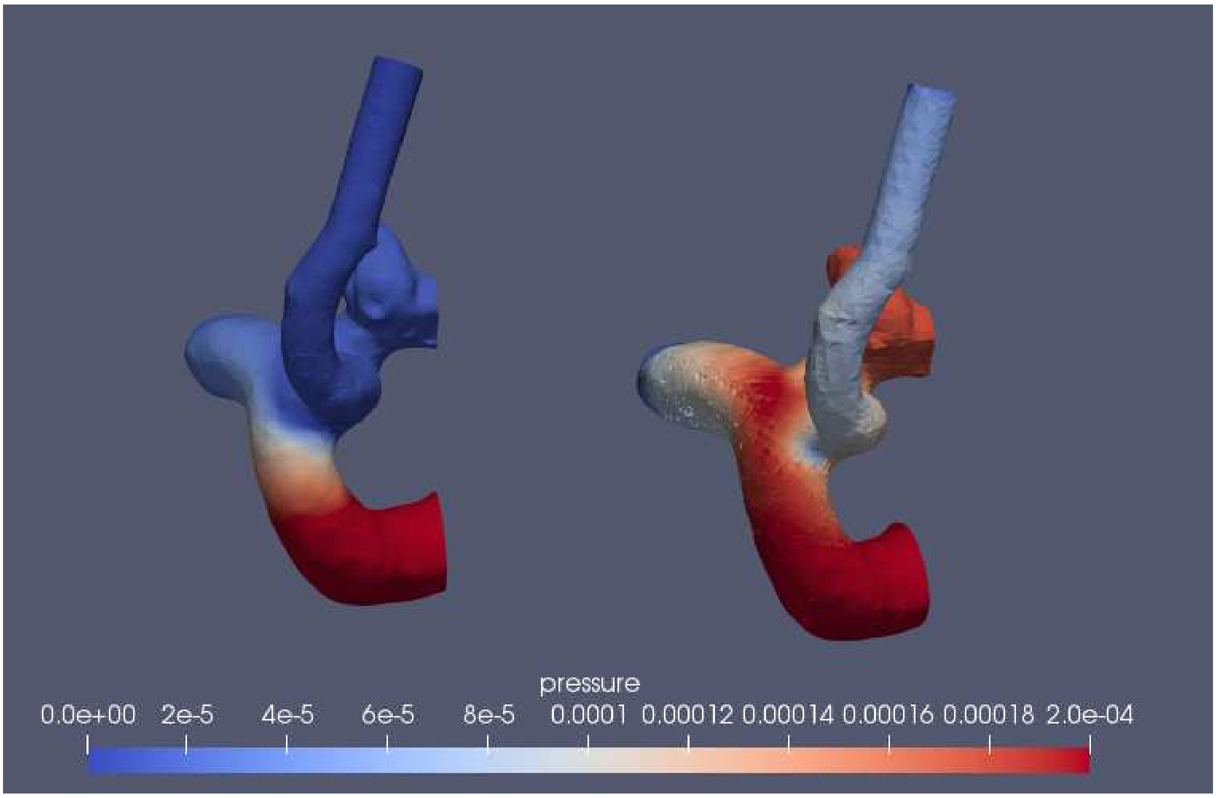

Figure 1: Relative pressure in Pa/ρ for simulated FD stent modeled as PM on the left, and modeled as no-slip boundaries on the right.

4 CONCLUSIONS

A solver capable of simulate heterogeneous PM was implemented. This solver has a good agreement with analytical solutions and a homogeneous solver of public domain. With a homogeneous PM in an ideal geometry, the solver presents same error as other solver with same refinement. Therefore, the source of this error must be caused by discretization criteria. The relative error in a PM with incrementing coefficients along the channel is not significant, verifying that the PM change its porosity along the channel.

With this solver, local porosity properties of FD stents can be taken into account. Raschi's model was used to represent a FD in a patient specific geometry. However, there are many PM models for FD stents in literature (Morales and Bonnefous, 2014; Raschi et al., 2014; Dazeo et al., 2017; Fernández et al., 2008). Many of them apt to provide different coefficients for each braided cell. This models can be now compared in real geometries.

The resulting implementation shows a good agreement with analytical and numerical methods. For the user, the description of a PM is straightforward.

ACKNOWLEDGEMENTS

This project is partly funded by PICT Start-up 2015-0006 and PICT 2016-0116 - FONCYT - ANPCYT of Argentina. N.D. is supported by CONICET PhD grant. The Titan Xp used for this research was donated by the NVIDIA Corporation. The financial support of these institutions is greatly appreciated.

REFERENCES

Dazeo N., Dottori J., Boroni G., Clausse A., and Larrabide I. Flow diverter stents simulation with cfd: porous media modelling. In *12th International Symposium on Medical Information Processing and Analysis*, pages 101601F–101601F. International Society for Optics and Photonics, 2017.

Fernández M.A., Gerbeau J.F., and Martin V. Numerical simulation of blood flows through a porous interface. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(6):961–990, 2008.

Jasak H., Jemcov A., Tukovic Z., et al. Openfoam: A c++ library for complex physics simulations. In *International workshop on coupled methods in numerical dynamics*, volume 1000, pages 1–20. IUC Dubrovnik, Croatia, 2007.

Kim M., Taulbee D.B., Tremmel M., and Meng H. Comparison of two stents in modifying cerebral aneurysm hemodynamics. *Annals of Biomedical Engineering*, 36(5):726–741, 2008.

Larrabide I., Aguilar M.L., Morales H.G., Geers A.J., Kulcsár Z., Rüfenacht D.A., and a. F. Frangi. Intra-aneurysmal pressure and flow changes induced by flow diverters: Relation to aneurysm size and shape. *American Journal of Neuroradiology*, 34(4):816–822, 2013.

Larrabide I., Kim M., Augsburger L., Villa-Uriol M.C., Rüfenacht D.A., and Frangi A.F. Fast virtual deployment of self-expandable stents: Method and in vitro evaluation for intracranial aneurysmal stenting. *Medical Image Analysis*, 16(3):721–730, 2012.

Ma D., Dargush G.F., Natarajan S.K., Levy E.I., Siddiqui A.H., and Meng H. Computer modeling of deployment and mechanical expansion of neurovascular flow diverter in patient-specific intracranial aneurysms. *Journal of biomechanics*, pages 1–8, 2012.

Morales H.G. and Bonnefous O. Modeling hemodynamics after flow diverter with a porous medium. pages 1324–1327, 2014.

Moukalled F., Mangani L., Darwish M., et al. The finite volume method in computational fluid dynamics. *An Advanced Introduction with OpenFOAM and Matlab*, pages 3–8, 2016.

Mut F., Raschi M., Scrivano E., Bleise C., Chudyk J., Ceratto R., Lylyk P., and Cebral J.R. Association between hemodynamic conditions and occlusion times after flow diversion in cerebral aneurysms. *Journal of neurointerventional surgery*, pages 1–5, 2014.

Raschi M., Mut F., Löhner R., and Cebral J. Strategy for modeling flow diverters in cerebral aneurysms as a porous medium. *International journal for numerical methods in biomedical engineering*, 30(9):909–925, 2014.

Sadasivan C.R., Cesar L., Seong J., Rakian A., Hao Q., Tio F.O., Wakhloo A.K., and Lieber

B.B. An original flow diversion device for the treatment of intracranial aneurysms: evaluation in the rabbit elastase-induced model. *Stroke; a journal of cerebral circulation*, 40(3):952–8, 2009.